Semi-supervised Adaptation of Assistant Based Speech Recognition Models for different Approach Areas

Matthias Kleinert[1], Hartmut Helmke[1], Gerald Siol[1], Heiko Ehr[1], Aneta Cerna[2], Christian Kern[3], Dietrich Klakow[4], Petr Motlicek[5], Youssef Oualil[4], Mittul Singh[4], Ajay Srinivasamurthy[5]
[1] Institute of Flight Guidance, German Aerospace Center (DLR), Braunschweig Germany,
[2] Air Navigation Services of the Czech Republic, Jenec, Czech Republic,
[3] Austro Control, Vienna, Austria
[4] Spoken Language Systems Group (LSV), Saarland University (UdS), Saarbrücken, Germany
[5] Idiap Research Institute, Martigny, Switzerland
firstname.lastname@{dlr.de, ans.cz, austrocontrol.at, lsv.uni-saarland.de, idiap.ch}

***Abstract*—Air Navigation Service Providers (ANSPs) replace paper flight strips through different digital solutions. The instructed commands from an air traffic controller (ATCos) are then available in computer readable form. However, those systems require manual controller inputs, i.e. ATCos' workload increases. The Active Listening Assistant (AcListant®) project has shown that Assistant Based Speech Recognition (ABSR) is a potential solution to reduce this additional workload. However, the development of an ABSR application for a specific target-domain usually requires a large amount of manually transcribed audio data in order to achieve task-sufficient recognition accuracies. MALORCA project developed an initial basic ABSR system and semi-automatically tailored its recognition models for both Prague and Vienna approaches by machine learning from automatically transcribed audio data. Command recognition error rates were reduced from 7.9% to under 0.6% for Prague and from 18.9% to 3.2% for Vienna.**

Keywords- Machine Learning, Assistant Based Speech Recognition, Unsupervised Learning, Command Prediction Model, Automatic Speech Recognition

I. INTRODUCTION

A. Problem

To ensure the acceptance of any new feature developed by an Air Traffic Management (ATM) project, it is imperative that its benefits are clearly recognizable for the end-user at the very beginning when they are confronted with new tools. Therefore, the deployment of decision and negotiation support tools in current ATM business still requires a strong and manual adaptation to the local environment to avoid end-user frustration. Total system costs can easily exceed the threshold of one million Euros. ATM system suppliers try to reduce costs by developing generic decision support tools, e.g. one basic Arrival Manager, which fits for many airports; an approach, which is often not successful and, therefore, costly adaptations are necessary after system installation or even worse the new tool is not used by the ATCo (air traffic controller).

Although air traffic control is an innovative business, paper flight strips are still in use. Written information on them is not available in digital form. Modern controller working positions (CWP), therefore, offer digital flight strips or totally stripless air traffic management systems. However, more than ever manual input is required from the controllers in such an environment. Others have the benefits and the controllers get additional workload. The Active Listening Assistant (AcListant®) project [1] has shown that Assistant Based Speech Recognition (ABSR) is a potential solution to reduce controllers' workload. The ABSR system developed by Saarland University (USAAR) and DLR analyses the controller pilot communication and shows the recognized commands in the radar label directly to the ATCo [2]. As command recognition rates better than 95% were achieved for Dusseldorf approach area, the controller only needs to manually correct the output of the speech recognizer in less than one of twenty cases [3]. The controller gets additional free cognitive resources, which increase safety. Furthermore AcListant® validated that fuel reductions of 60 liters per aircraft (based on an A320) and up to two landings more per hour are possible [4].

For Dusseldorf all the ABSR models were manually developed and maintained. This approach is too expensive if this manual work is required again for any new airport.

B. Solution

The Horizon 2020 SESAR project MALORCA (Machine Learning of Speech Recognition Models for Controller Assistance) offers machine learning (ML) framework as a general, cheap and effective solution to automate the adaptation and customization process of ATM decision support tools [5]. Adaptation of speech recognition models were selected as a first show-case of MALORCA. The MALORCA consortium consists of two members from academia, Saarland University (Germany) and Idiap Research Institute (Switzerland), Air Navigation Service Providers from Czech Republic (ANS CR) and Austria (Austro Control) representing the user needs, and

 The work is partly funded by SESAR Joint Undertaking (Grant Number 698824). Youssef Oualil contributed to this paper during his work period at Saarland University. Ajay Srinivasamurthy is currently with Amazon.com, Bangalore, India and contributed to this paper during his work period at Idiap Research Institute before joining Amazon

the project lead German Aerospace Center (DLR) as the connecting element between basic research and business needs.

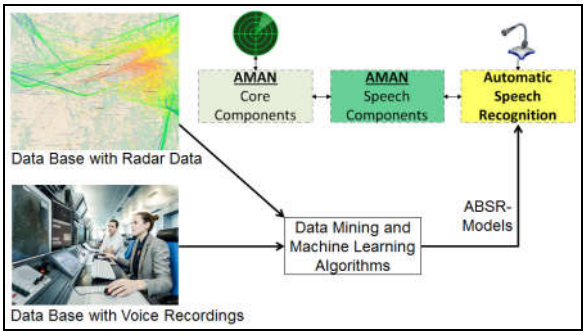

Figure 1 Principal idea of MALORCA project to learn from historic data

The proposed solution builds on the huge amount of target data recorded every day in the operation rooms (Figure 1). Each Air Navigation Service Provider generates Mega Bytes or even Giga Bytes of radar data and voice recordings on a daily basis. These recordings are the input for machine learning algorithms, which improve the models of a basic ABSR system. Improvement is possible once or permanently on daily basis. If a new waypoint is added it would be learned, if a waypoint is removed it would be "unlearned" etc.

C. *Paper Structure*

In the next section we present related work with respect to machine learning and speech recognition applications in ATM. Section III describes the building blocks including the different recognition models of an Assistant Based Speech Recognizer. In section IV we show the independent training of each ABSR model, whereas section V shows the iterative and dependent improvement of each of the three main ABSR models. The last section VI before the conclusions presents the results of the iterative model improvements.

II. Related Work

A. *Speech Recognition Applications in ATM*

Artificial intelligence (AI) and in particular machine learning applications have made a significant progress in the last few years, enabling computers to make a series of major breakthroughs that were previously impossible. One of the successful fields is automatic speech recognition (ASR), which has shown remarkable improvements in understanding human conversational speech.

Speech Recognition applications have dramatically improved during the last decade (e.g. Siri®, Alexa, Google Assistant). The integration of ASR in ATM training started already in the late 80s [6]. Today ASR applications go beyond simulation and training. ASR is e.g. used to get more objective feedback of controllers' workload [7]. Chen and Kopald used speech recognition to build a safety net for airport surface traffic to avoid aircraft entering a closed runway [8]. Most recently they presented an approach to detect pilot read back errors [9].

A good introduction into the state-of the art of ASR applications in the ATM domain until 2014 is given by Nyuyen and Holone [10]. They identified five challenges that need to be overcome, so that ASR applications are more successful in the ATM domain:

1. Callsign Detection: many different pronunciations and word combinations exists e.g. already for the callsign DLH123A: lufthansa one two three alpha, or hansa three alpha or delta lima hotel one two three alpha …
2. Poor Input Signal Quality: background noise, 8 kHz or even worse signal frequency, late pressing of push-to-talk button, hesitations, slurred speech etc.
3. Ambiguity: Although the vocabulary in controller pilot communication is quite limited and phraseology is restricted, recognition rates are still far from being perfect: two four five could be interpreted as a callsign, a heading, a speed or a flight level.
4. Use of Non-Standard Phraseology: Nguyen et al. [10] claim with respect to [11] that at least 80% of all pilot transmissions contain at least one error. We will not be able to change the controllers and pilots. The solution is to adapt the ASR systems. Usage of standard phraseology will improve bit by bit, when pilots and controllers recognize that they are directly benefitting from using correct phraseology.
5. Dialect, Accents and Multiple Languages: Spanish controllers may speak to a VFR flight in Spanish and to a lufthansa callsign in English or the greeting to the lufthansa maybe in German whereas the rest is in English.

One promising approach to improve ASR performance is using context knowledge regarding expected utterances. These attempts go back to the 80s [12], [13]. Context may heavily reduce the search space and lead to fewer misrecognitions [14]. DLR and Saarland University went one step further when using an Arrival Manager (AMAN) as context source [15], [16].

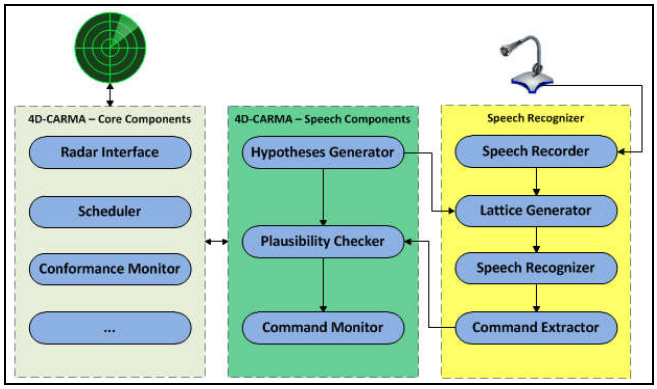

Figure 2 Components of ABSR; in green components of AMAN 4D-CARMA; in yellow components of core speech recognizer (taken from [19])

An Arrival Manager (AMAN) (4D-CARMA) as seen in Figure 2 analyzes the current airspace situation and predicts

possible future states, which are used by the "Hypotheses Generator" to predict a set of possible commands [2]. This significantly reduces the search of the "Lattice Generator" [17], [18]. The search lattice is dynamically regenerated and contains a search tree for all possible word sequences determined by the "Hypotheses Generator". The "Speech Recognizer" finds the most probable path in the search tree. The output of the "Command Extractor" is checked again by the "Plausibility Checker", determining whether the recognized commands are reasonable in the current situation. The "Command Monitor" analyzes the future behavior of the aircraft (via radar data), whether they are in line with the "Command Extractor's" output.

B. *Supervised and unsupervised learning*

Machine learning aims to model rules that can map input data to meaningful output labels. Input data can be position, speed and altitude of an aircraft, output label could the possible command type (DESCEND, REDUCE, TURN_LEFT). Output labels can be categorical, in which the task is called classification, or could be continuous valued in a regression task. Current machine learning methods require examples to train such models.

Based on the training data available, machine learning methods can be supervised, unsupervised or semi-supervised [20], [21]. In supervised learning, we require data samples and corresponding output labels, and several different algorithms can be used to learn the input output relationship. However, in unsupervised learning, the output labels are not available and the machine learning algorithm just uses the data examples to learn both output labels and the rules to model data. Typical unsupervised learning approaches include data clustering to partition data according an optimization criterion. In semi-supervised learning, partially labeled set of examples are used to build a machine learning model.

While supervised learning requires expensive labeled data, unsupervised learning methods often suffer from poor performance. A good tradeoff is semi-supervised learning, where an initial seed model is built using limited amounts of labeled examples, which is then improved further using a large number of relatively cheap unlabeled examples.

Several different machine learning tools for classification and regression exist. Recently, neural network models have been shown to accurately learn arbitrary input output relationships. Neural network models require extensive computational resources and are mainly effective when large number of examples are available, e.g. to build acoustic models for the ABSR system [22]. In this paper, we explore semi-supervised learning and adaptation of different ABSR components.

Specifically, for certain ABSR components like acoustic and language models (presented in section III), semi-supervised training has been beneficial in improving performance in low-resource scenarios, (i.e. small amount of training data as in MALORCA) [23], [24], [25], [26]. For acoustic modeling, researchers have applied various data-selection schemes to utilize the additional unlabeled data [27], [28], [29], [30], [31], [32]. In this paper, we apply a technique built specifically to account for semantics of the ATM domain [27]. For language modeling, however, the additional data available can still be unreliable and we automatically generated further transcripts which are used to train language models. Prior work [33], [34] has applied these techniques successfully for speech recognition in other domains different from ATM.

C. *ABSR combined with CPDLC*

A new method that is increasingly used to transfer messages from the ground to the cockpits is datalink technology. CPDLC (Controller Pilot Data Link Communications) [35] is a standard to formalize command and information exchange between controllers and pilots and vice versa. Today in Austro Control and ANS CR this system is mainly used with aircraft in the enroute phase of flight and for a limited type of commands. At least maximum transmission times are longer than those of voice communications [36]. Hence, voice is and will remain the most important means of communication in ATM.

Nevertheless, ABSR is expected to be also beneficial if CPDLC is used. Even if not talking to pilots the controllers may still use voice to do the required inputs into the system and the transmission of the appropriate datalink-messages will automatically be triggered by the system. Human machine interfaces using a comprehensive speech recognizer are proven to be highly efficient so it is expected to provide benefits in any currently foreseeable technological environment in ATM.

III. BUILDING BLOCKS OF ASSISTANT BASED SPEECH RECOGNITION

Assistant Based Speech Recognition (ABSR) normally uses three main models (dark blue ovals on left side in Figure 3), which need to be trained / adapted for each ATM environment (approach area) separately:

1. Acoustic Model,
2. Language Model (e.g. grammar) and
3. Command Prediction Model (CPM).

Figure 3 shows in the upper part how those three models are used within ABSR: Rectangles describe tasks and light blue ovals inputs and outputs. The CPM is used by the Hypotheses Generator to derive a set of commands (Command Hypotheses), which are possible in the current situation. These commands are used as input for Automatic Speech Recognition (ASR) to reduce the search space size and to guide the search process of the speech recognition system.

The other two models (acoustic and language) are directly used by ASR. A controller utterance, given in form of audio signal, is transformed into a feature vector X. Acoustic and language model are used to transform the feature vector into a sequence of spoken words $W = (W_1, W_2, W_3, \ldots)$, i.e. we apply the Bayes' theorem to find the word sequence which maximize a posteriori probability $P(W_1, W_2, W_3, \ldots / X)$. In particular,

statistical language models have shown to be beneficial in comparison to a hand-written grammar in an ATM environment [37]. This automatic modeling scheme is simple to implement depending only on the availability of the transcribed text and easy to manage. A sequence labeling approach (Command Extractor in Figure 2) then extracts the relevant concepts and commands from the recognized word sequence W by also using the language model and especially the Command Hypotheses.

An ontology for modelling the semantics of controller word sequences is being developed by SESAR 2020 project PJ 16-04 [38]. It describes also the allowed types (e.g. DESCEND, REDUCE, INCREASE, CLEARED_ILS).

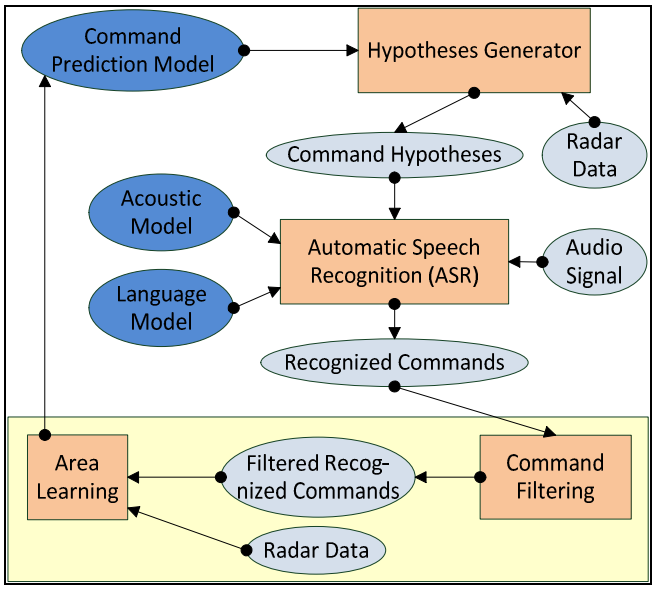

Figure 3 Interaction of Acoustic, Language and Command Prediction Model

IV. Independent Training of the Different Models

This section describes the training of the three main models of ABSR. First we introduce the used metrics to determine the quality of our models. In the next subsection we concentrate on the training of the CPM. The last two subsections describe the training of the acoustic and language model. In the following section we then describe the iterative training of all three models.

A. Used Metrics

The following metrics are used to determine the quality of the learned models:

- Word Error Rate (**WER**): number of words which are wrongly recognized (substitutions) plus number of words not pronounced, but recognized (insertions), plus number of words not recognized, but pronounced (deletions), divided by the total number of spoken words
- Total number of given commands (**#TgC**),
- Command recognition rate (**RR**): number of correctly recognized commands, which are not rejected by CPM, divided by #TgC (a command is correct if both the callsign and the command type and the command value are correctly recognized) ,
- Command recognition error rate (**ER**): number of recognized commands which were not spoken and not rejected, divided by #**TgC**,
- Pure command recognition rate (**PRR**): number of correctly recognized commands, without considering rejection by Command Filtering using CPM, divided by #TgC,
- Pure command recognition error rate (**PER**): number of recognized commands which were not spoken (false recognitions), divided by #**TgC**,
- Command prediction error rate (**CpER**): number of commands included in gold commands, which were not predicted, divided by #**TgC**,
- Average number of predicted commands per aircraft (**#NPC**),

We reject a recognized command if it is not predicted by CPM. PRR and PER consider the output of the pure ASR system without using the set of predicted commands. Rejections are also possible here, i.e. if the output callsign of ASR is NO_CALLSIGN resp. if the output type of the command is NO_CONCEPT (see [19] for a more detailed definition of these rates). Sometimes more commands are erroneously recognized than given (so called insertions). This may result in an increase of **ER** resp. PER, if not rejected. If fewer commands are recognized than given (so called deletion), this is always counted as a rejection.

B. Training of the Command Prediction Models

For each command type a prediction area is modelled as shown by the dark hash symbols ('#') in Figure 4. Additionally we add a set of predefined rules to each command type (e.g. IF flight type is arrival AND controller working position is Feeder AND speed > 220 knots). If the "Hypotheses Generator" detects that a lat/long position of an aircraft is inside an area of a specific command type and the rule condition for this area is true, the command values related to that flight and command type are predicted for that aircraft. Each symbol in the prediction area (see Figure 4) represents a square of 1 nm by 1 nm. These areas can be created manually [39] or learned automatically from transcribed controller utterances and corresponding recorded radar data. This, however, requires either expert knowledge for manual creation and/or expensive manual transcription work of recorded utterances. In order to remove the need of manual work, our approach tries to learn these areas from automatic transcriptions (task "Area Learning" in Figure 3). For each controller utterance the corresponding lat/long positions are known from the recorded radar data, but the (correct) controller commands, however, are unknown. The only things we know are the recognized commands from the Automatic Speech Recognition in Figure 3.

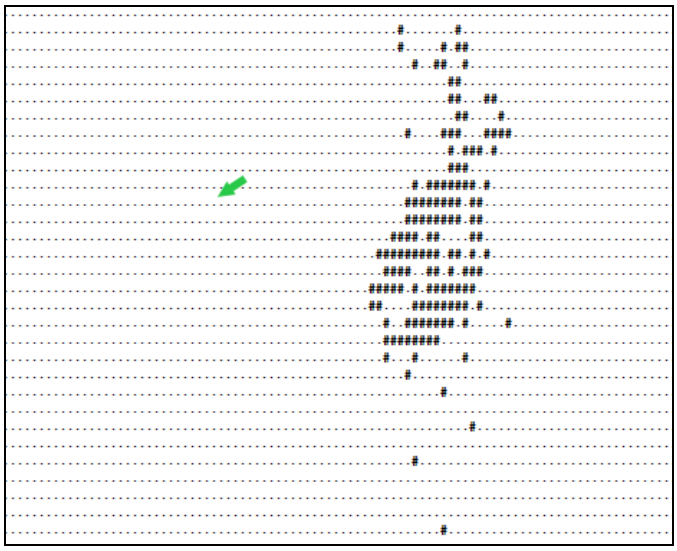

Figure 4 Prediction area of CPM for Cleared ILS-Command for Arrivals, each dot represents an area of one square mile; the small green arrow represents the position of runway 24.

If we have a controller utterance like, "sky_travel two five zero nine reduce two one zero knots", ABSR should normally recognize the expected command "TVS2509 REDUCE 210". Afterwards this command could be used, together with the corresponding radar data (which amongst others includes flight plan information) for automatic learning of the command prediction model (CPM).

The Command Filtering in Figure 3 tries to filter out false recognitions. A resulting command prediction area for the CLEARED_ILS command for arrivals for the director position in Prague is shown in Figure 4.

With a closer look to Figure 4 two problems become obvious: (1) outliers, which are probably the result of false recognitions the "Command Filtering" did not catch, (2) small gaps, in which no Cleared ILS command was observed in the training data, but is very likely if more data would be available.

To close the gaps and also expand/smooth the borders of the learned areas we assumed that a valid command that appears at a certain position in the training data is not only valid for this position, but also for the nearby positions. That means, we do not only mark the respective 1*1 area, in which a command occurs, but also the surrounding areas. In this context an expansion window size of 13 means that we also mark the 168 neighbors (13*13-1) of a certain lat/long position in which a command occurred. In addition to the window size 25 in Figure 5 we experimented with different window sizes for expansion and gap closing. More details also with respect to filtering of outlier are provided in [40].

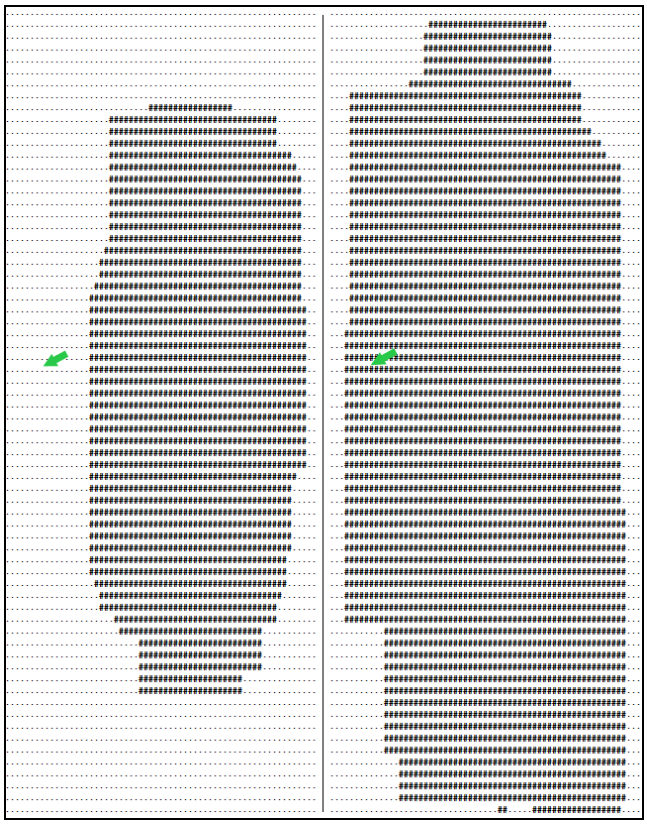

Figure 5 Prediction area of CPM for Cleared ILS-Command (left) and HANDOVER_FREQUENCY (right) for Prague Arrivals for Director (expansion window 25x25); green arrow is runway 24

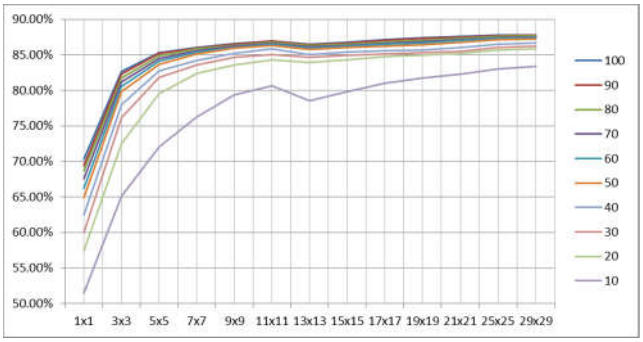

Figure 6 Dependency of Command Recognition Rate (RR) on training data size and window size for Prague for all command types

Figure 6 and Figure 7 show the dependency of Command Recognition Rate (RR) and Command Recognition Error Rate (ER) from window size and amount of used training data. It is not surprising that the RR increases with larger window sizes and with the amount of training data used, because more commands are predicted by the Hypotheses Generator. On the other hand ER also increases with increasing window size because less false recognitions are rejected since they were also predicted by CPM.

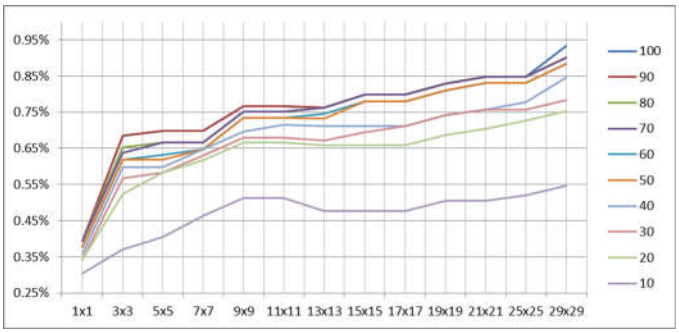

Figure 7 Dependency of Command Recognition Error Rate (ER) on training data size and window size for Prague for all command types

All in all the best compromise between high RR and low ER was observed with a window size of 25 [40]. The resulting areas of the CLEARED_ILS and HANDOVER_ FREQUENCY command type are shown in Figure 5.

C. *Training of the Acoustic Model*

To develop the acoustic model, we rely on open source out-of-domain English corpora used to initialize the training [33]. As in-domain ATM recordings from both Prague and Vienna approaches are of 8 kHz quality, the same type of data is used over all acoustic modeling.

A lexicon is one of the essential blocks of an automatic speech recognition system. To be able to model all possible commands spoken by ATCos, we expand standard CMU-Sphinx dictionary [41] by all ATM in-domain words (such as airline names, waypoint names, etc. for a given approach area) to form an extended pronunciation lexicon subsequently used by acoustic and also language models in the ASR engine.

For acoustic modeling, we rely on conventional technology combining deep learning (i.e. deep neural networks) employed in Hidden Markov Modelling (HMM) framework. The technology referred to as hybrid acoustic modeling not only offers state-of-the-art performance, but also allows for rapid acoustic domain adaptation, which is essential for our ABSR system and used for:

- speaker-dependent modeling: As we know which controller is speaking the general acoustic model can be adapted to each ATM controller to model speaker-specific variability captured in speech resulting in a reduction of the word error rate (WER),
- bootstrapping the model from rich resources (i.e. out-of-domain dataset) leveraging other ASR application domains and adapting the generic model to a target-domain: As MALORCA does not offer sufficient amount of training data to develop robust acoustic models for ASR, we used 150 hours generic manually transcribed open-source English speech data (e.g. meeting recordings or read-speech corpora). The initial acoustic model is eventually adapted using in-domain dataset for the target ATM approach.
- iterative re-training: Acoustic, language and command prediction models provide confidence measures which can be used to assess quality of automatically generated transcripts related to new speech data. The fused confidence measure can be directly applied to select the relevant speech data and iteratively re-train the hybrid acoustic model.

D. *Training of the Language Model*

Language modeling techniques like the grammar-based models provide a large set of rules to cover the phraseology used by controllers whereas, the statistical language models learn these rules automatically along with the deviations regularly made by controllers (assuming enough training data is available) and also adapt to these deviations in a more robust manner than a grammar-based model.

TABLE 1: GRAMMAR MAPPING THE COMMAND "DLH23B REDUCE 250" TO CONCEPTS (CLASSES)

Command	Mapped Concept
DLH	AIRLINE Identifier
23	CALLSIGN NUMBER
B	CALLSIGN LETTER
REDUCE	ACTION
250	NUMBER

In our experimental work, we continue exploring statistical language models. Even though, the statistical language models have been shown to perform better than Grammar-based models [37], the MALORCA project has raised a unique challenge when building these statistical models, as the initial amount of transcribed data is relatively small (< 4 hours). As this can lead to a poor coverage of ATM commands, we alleviate this problem in MALORCA, by leveraging the ICAO grammar [42] and constructing a hybrid statistical language model from this grammar and already trained, statistical language model.

The grammar specifies the set of rules, defining the correspondence of command words to concepts (classes). An example of this correspondence applied to the command "DLH23B REDUCE 250" is shown in Table 1.

These classes can then be used to build a class-based statistical language model [43] which has shown improved performance in a speech recognition system. Intuitively, this class-based language model allows overcoming the lack of data by mapping everything to a class space. In this class space, correlations can be learned at a concept level, unlike the regular statistical language models used earlier [18].

These class-based and regular statistical language models are linearly interpolated **Fehler! Verweisquelle konnte nicht gefunden werden.** to produce the hybrid statistical language model. Finally, this hybrid language model is converted to a first-pass decoding finite state transducer [21] and employed in the ABSR pipeline.

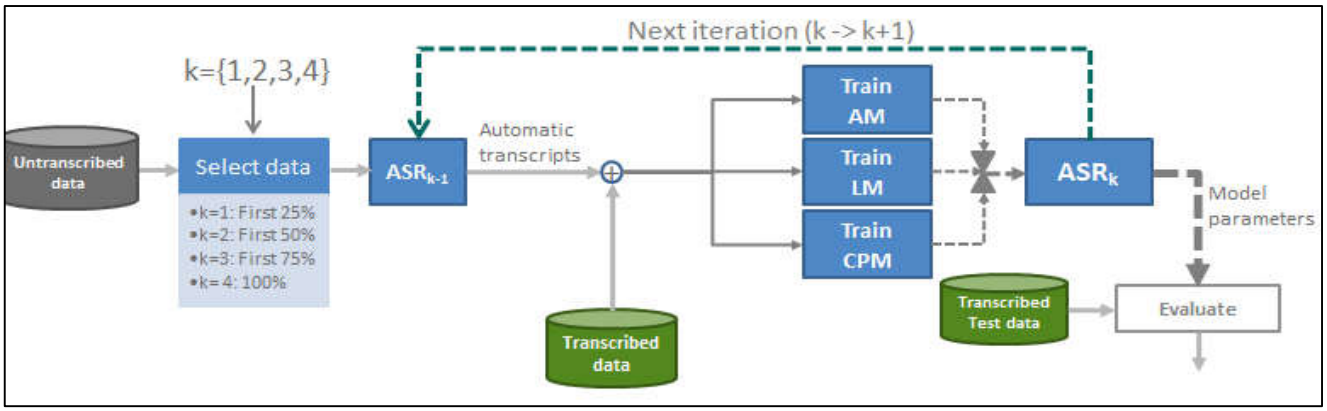

Figure 8 Detailed setup for iterative improvement of recognition models

The data for training the different speech recognition models is produced iteratively. Hence, the language models are re-trained each time a new portion of data is automatically transcribed. This re-training involves creating the regular statistical language model on the combined original and automatically-transcribed portion. Similarly, the class-based statistical language model is re-trained on this combined dataset. See also **Fehler! Verweisquelle konnte nicht gefunden werden.** for details of acoustic and language model training in MALORCA project.

V. Iterative Model Improvement

As shown in the bottom yellow shaded part of Figure 3, automatic learning of the predictions areas will result in an improved "Command Prediction Model", which we expect will also improve the "Command Hypotheses" iteratively resulting in better "Recognized commands". The aim of the MALORCA project, however, is to learn/improve also the other ABSR models. The "Command Filtering" in Figure 3 helps also to improve both the acoustic and language model, because the learning algorithms for acoustic and language model use the feedback from the additional sensor "Radar data" to decide whether an automatic transcription is good or improvable.

Figure 9 shows the principal setup. First we use the untranscribed data from August 2016 and train all three models from them. This step results in an ABSR System called ABSR August. We use this system and evaluate with all the testing data and determine the metrics defined in subsection IV.A for the ABSR August system. Then we additionally add untranscribed data from September and retrain our models on the data of August and September. We get a system ABSR September and determine the metrics again. In the same way we continue with the data from the other months. In detail we do not always use the data from exactly one month, but rather we select the data according to the total length of the dataset: practically, we first select 25%, then the first 50%, then the first 75% and then all (100%) data. In fact, this is nearly equivalent to taking the data on monthly basis.

Figure 8 shows our approach in more detail. We start with "zero" amount of (0%) untranscribed data and add 70% of the transcribed data and we train both the acoustic and language model (AM, LM) of the ASR system. For the command prediction model (CPM) we use 10% of the untranscribed data, but we completely exclude transcribed data to demonstrate that CPM can be learned from untranscribed data only. With this baseline model (model-0%) we evaluate the command recognition, command recognition error rates, etc.

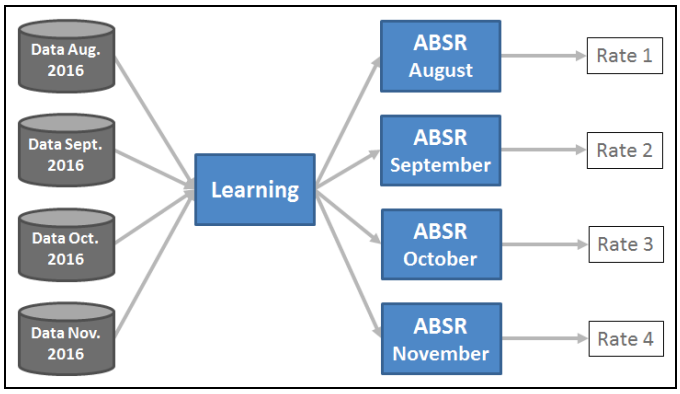

Figure 9 General setup for iterative improvement of recognition models

The baseline recognizer is then improved through multiple iterations with more and more data. At the beginning it is used to transcribe the first 25% of the untranscribed controller utterances and the baseline CPM is applied to classify all the recognized commands into "good" and "bad" learning examples. Generated command recognitions which are not part of the predicted command hypotheses are classified as bad examples and not used for acoustic and language model training (see Table 2 for amount of data pre-selected for re-training).

TABLE 2: AMOUNT OF DATA PRE-SELECTED FROM AUTOMATICALLY TRANSCRIBED DATA OVER (K) ITERATIONS

	Prague		Vienna	
k (split)	**Total (hours)**	**% selected**	**Total (hours)**	**% selected**
1 (25%)	4.5	56.4	4.6	48.7
2 (50%)	9.2	75.8	9.1	73.6
3 (75%)	13.7	78.0	13.7	77.2
4 (100%)	18.3	78.4	18.2	78.8

The resulting model of the AM and LM is used afterwards to automatically transcribe again the untranscribed 25% data set. In this second step the recognized commands are used to create the improved CPM. Eventually, ASR1 system (based on 25% of the data) is obtained. This system is then used as a starting point for the next iteration (applying 50% of the untranscribed data). This process is repeated until all of the data has been used for training of AM, LM and CPM.

VI. Results of Iterative Model Improvement

For evaluation we used different test utterances from Prague and Vienna approach which were manually transcribed (see Table 3), i.e. for these utterances the correct transcription (so called gold commands) were known. For Prague we have 3'039 different utterance from 31 different sessions, i.e. recording periods lasting between 30 minutes and two hours. Command transcription resulted in 5'339 different commands (i.e. an utterance contains on average 1.8 commands)

TABLE 3: SIZE OF TEST DATA SET

Approach Area	# Utterances	# given commands	# sessions
Prague	3'039	5'339	31
Vienna	3'012	4'211	24

A. Results for Vienna Approach

TABLE 4: METRICS FOR ITERATIVE IMPROVEMENT FOR VIENNA APPROACH

Amount of untranscribed data used	RR [%]	ER [%]	PRR [%]	PER [%]	CpER [%]	#NPC
0%	60.0	1.6	67.2	18.9	15.2	14
25%	80.2	3.5	84.0	7.4	6.7	29
50%	82.4	2.8	84.7	6.7	4.6	39
75%	84.2	3.0	85.6	7.0	3.5	47
100%	85.2	3.2	86.4	6.6	3.2	53

Results based on 4211 given commands from 3.84 hours of speech excluding silence, i.e. 21.1 hours of radar data time

In Table 4 the results for Vienna Approach are presented. The first row (0%) in this table shows the results when we first trained the three models (AM, LM, CPM) without any untranscribed data for AM and LM training, although 10% of the untranscribed data was already used for the CPM (for more details see section V). The following rows show the results for the training with 25%/50%/75% and 100% of the untranscribed data for all of the three models.

The first two columns of the table (RR and ER) show the performance of the complete ABSR system. It includes command filtering with the help of the CPM. The columns PRR and PER shows the performance of the pure ASR system without any command filtering. There is of course always a decrease in recognition rate when using command filtering, i.e. from PRR to RR, because sometimes the command filtering through the CPM falsely rejects correctly recognized commands. With the 0% system for example there is a decrease in recognition rate by 7.2% from pure ASR to ABSR system, but the use of the CPM from the ABSR system decreases the error rate at the same time significantly by 17.3%.

The last two columns of Table 4 give some additional information about the quality and the amount of predicted commands per aircraft (for more detail see subsection IV.A).

B. Results for Prague Approach

Table 5 presents the same metrics as Table 4, but for the Prague approach area. We can see a similar increase in RR with more data for training and a similar influence of the ABSR system on ER. Overall the numbers for Prague are better than for the Vienna approach. The main reason for that is the better quality of audio data provided for Prague approach.

TABLE 5: METRICS FOR ITERATIVE IMPROVEMENT FOR PRAGUE APPROACH

Amount of untranscribed data used	RR [%]	ER [%]	PRR [%]	PER [%]	CpER [%]	#NPC
0%	79.8	0.29	85.9	7.90	8.1	28
25%	90.2	0.32	93.7	2.2	4.4	45
50%	91.3	0.37	93.5	2.3	3.0	58
75%	91.7	0.45	93.6	2.4	2.5	67
100%	91.9	0.60	93.7	2.4	2.3	70

Results based on 5339 given commands from 4.69 hours of speech excluding silence, i.e. 25.7 hours of radar data time

If we look at the improvements in RR from 75% to 100%, it seems relatively small compared to the loss in ER. However, the November 2016 data that was used for the 100% evaluation contained a frequency change for different controller working positions in Prague. This change had an impact on the precision of the ABSR system. Deeper analysis with more data recorded after November 2016 would be necessary to determine the extent of this impact.

C. Interpretation of Results

The results in Table 4 show that the learning curve of Vienna does not reach its saturation limits. Increasing the data size by a factor of 2 (from 25% to 50% and from 50% to 100%) still improves the values. RR increases by 2.2% (absolute) from 25% data size to 50% data size and again by 2.8% (absolute) from 50% to 100%. If we extrapolate the currently available 100% data by a factor of 8 RR of 90.2 % with ER of 4.4% for Vienna seems to be possible.

The results in Table 5 show that for Prague data it seems like the trained models are already close to saturation, , i.e. the currently available 100% of learning data would be already sufficient. But as already mentioned in the previous subsection the change of frequencies that occurred in November 2016 in Prague would require further analysis to determine its influence on the overall performance. If we perform the same data extrapolation as for Vienna with 8 times more data the recognition rate for Prague could reach 92.6% with an error rate of 1.3%.

If we compare the command prediction error rate (CpER) the models for Prague seem to be better than for Vienna (1.8% versus. 2.3%). There are different reasons for this:

- Prague radar data covers usually 60 to 90 minutes of runtime, whereas Vienna data sometimes covers more than 14 hours, i.e. a callsign which is landing at Vienna may start in the same data set. The (simplified) ARR/DEP classification of a callsign, however, is kept for the whole day. This miss classification leads to wrong predictions for some aircraft. To reduce this influence we manually corrected some of the miss classifications.
- In Vienna only runway 34 for ARR and runway 29 for DEP were modelled. In some of the provided data, however, both runways were used for inbounds. Commands for inbound aircraft on runway 29 were, therefore, often wrongly predicted. To reduce this influence we excluded all directories from evaluation which contained a significant number of inbounds at runway 29

Table 4 and Table 5 show that the command recognition error rate (ER) goes up, when the amount of training data is increased. The command recognition rate (RR) then also goes up. The same applies for the number of predicted commands per aircraft (see column #NPC). This, however, increases the search space of possible command hypotheses. The CPM performance of differentiating between good and not so good (or not so likely) command predictions decreases. In technical terms, the recall (incorrectly rejected commands) in context increases while the precision (number of falsely accepted commands) decreases. Increasing the number of training data should not increase the number of predicted commands. This means that the command prediction model learning needs to be improved, to make better predictions. This is e.g. possible by adding plausibility value to each predicted command and limiting e.g. the set of predicted commands to the N (e.g. 50) most plausible one for a given aircraft.

VII. CONCLUSIONS

MALORCA started with the idea that radar data as an additional sensor modality will improve unsupervised learning of speech recognition models. The approach was successfully validated using historic radar and speech recordings from the ops rooms of Vienna and Prague. Speech signal quality significantly influences the ABSR system performance. This is clearly demonstrated by the difference in achieved command recognition error rates which were below 0.6% for Prague and restricted to 3.2% for Vienna having a much higher signal to noise ratio. We also demonstrated by incrementing the amount of training data in steps of 25% that an iterative learning is possible, i.e. the recognition rate will further improve by adding more and more training data.

We have shown that even with small amounts of transcription data (in combination with large amount of out-of-domain data) we can achieve command recognition rates close-to 92%. MALORCA is based on four hours of transcribed and 21 hours of untranscribed speech data for both Prague and Vienna. In terms of human effort, developed machine learning algorithms have significantly brought down the transcription effort. Nevertheless manual effort for pre-processing the radar data is still needed which should be reduced if learning is implemented directly in the ops-room from thousands of hours. This result together with the easy adaptable basic ABSR system for approach control will be the key to developing and deploying ABSR to different approach areas. Overall, the impact of the solutions of the MALORCA project when integrated into the current ATM procedures is expected to be high, especially due to minimizing the total costs related to the implementation of decision and negotiation support systems and related to the maintenance and system changes towards new ATM procedures. Next steps are on the one hand improvement of model learning (acoustic, language and command prediction model) by e.g. using plausibility values for acoustic and command prediction output. On the other hand MALORCA has shown that machine learning already now eases implementation and adaptation of ABSR systems to different deployment areas. MALORCA's learning approach has to leave the laboratory environment and needs to be applied in the ops room by both learning on daily basis and also by also at hub airports.

ACKNOWLEDGMENT

We would like to thank all the controllers who anonymously provided us with real world command examples and also our MALORCA partners from Austro Control and from Air Navigation Service Provider of Czech Republic.

REFERENCES

[1] The project AcListant® (Active Listening Assistant) http://www.aclistant.de/wp, n.d.

[2] H. Helmke, H. Ehr, M. Kleinert, F. Faubel, and D. Klakow, "Increased acceptance of controller assistance by automatic speech recognition," in 10th USA/Europe Air Traffic Management Research and Development Seminar (ATM2013), Chicago, IL, USA, 2013.

[3] H. Helmke, O. Ohneiser, Th. Mühlhausen, M. Wies,."Reducing controller workload with automatic speech recognition", in IEEE/AIAA 35th Digital Avionics Systems Conference (DASC). Sacramento, California, 2016.

[4] H. Helmke, O. Ohneiser, J. Buxbaum, Chr. Kern, "Increasing ATM efficiency with assistant-based speech recognition", in 12th USA/Europe Air Traffic Management Research and Development Seminar (ATM2017). Seattle, Washington, 2017.

[5] MALORCA: Machine Learning of Recognition Models for Controller Assistance, Homepage, www.malorca-project.de, n.d.

[6] C. Hamel, D. Kotick, and M. Layton, "Microcomputer system integration for air control training," Special Report SR89-01, Naval Training Systems Center, Orlando, FL, USA, 1989.

[7] J.M. Cordero, N. Rodríguez, J.M. de Pablo, and M. Dorado, "Automated speech recognition in controller communications applied to workload measurement," 3rd SESAR Innovation Days, Stockholm, Sweden, 2013.

[8] S. Chen, H.D. Kopald, A. Elessawy, Z. Levonian, and R.M. Tarakan, "Speech inputs to surface safety logic systems," IEEE/AIAA 34th Digital Avionics Systems Conference (DASC), Prague, Czech Republic, 2015.

[9] S. Chen, H.D. Kopald, R. Chong, Y. Wei, and Z. Levonian, "Read back error detection using automatic speech recognition," 12th USA/ Europe Air Traffic Management Research and Development Seminar (ATM2017), Seattle, WA, USA, 2017.

[10] V. N. Nguyen, H Holone, "Possibilities, challenges and the state of the art of automatic speech recognition in Air Traffic Control," International Journal of Computer and Information Engineering, Vol 9, No. 8, 2015, pp.1940-1949.

[11] C. M- Geacar, “Reducing pilot/ATC communication errors using voice recognition,” in Proc. of ICAS, Vol 2010, 2010.

[12] S.R. Young, W.H. Ward, and A.G. Hauptmann, “Layering predictions: Flexible use of dialog expectation in speech recognition,” in Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI89), Morgan Kaufmann, 1989, pp. 1543-1549.

[13] S.R. Young, A.G. Hauptmann, W.H. Ward, E.T. Smith, and P. Werner, “High level knowledge sources in usable speech recognition systems,” in Commun. ACM, vol. 32, no. 2, Feb. 1989, pp. 183-194.

[14] D. Schäfer, “Context-sensitive speech recognition in the air traffic control simulation,” Eurocontrol EEC Note No. 02/2001 and PhD Thesis of the University of Armed Forces, Munich, 2001.

[15] T. Shore, “Knowledge-based word lattice re-scoring in a dynamic context,” Master Thesis, Saarland University (UdS), 2011.

[16] T. Shore, F. Faubel, H. Helmke, and D. Klakow, “Knowledge-based word lattice rescoring in a dynamic context,“ Interspeech 2012, Sep. 2012, Portland, Oregon.

[17] A. Schmidt, “Integrating situational context information into an online ASR system for Air Traffic Control,” Master Thesis, Saarland University (UdS), 2014.

[18] Y. Oualil, M. Schulder, H. Helmke, A. Schmidt, and D. Klakow, “Real-time integration of dynamic context information for improving automatic speech recognition,” Interspeech, Dresden, Germany, 2015.

[19] H. Helmke, J. Rataj, T. Mühlhausen, O. Ohneiser, H. Ehr, M. Kleinert, Y. Oualil, and M. Schulder, “Assistant-based speech recognition for ATM applications”, in 11th USA/Europe Air Traffic Management Research and Development Seminar (ATM2015), Lisbon, Portugal, 2015.

[20] R. O. Duda, P. E. Hart, D. G. Stork, “Unsupervised learning and clustering,” Pattern classification (2nd ed.), Wiley, 2001.

[21] T. Hastie, R. Tibshirani, J. Friedman, “The elements of statistical learning: data mining, inference, and prediction,” New York: Springer, 2009, pp. 485–586.

[22] G. Hinton, L. Deng, D. Yu, G. Dah et al., “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, ” IEEE Signal Processing Magazine, Vol. 29, pp. 82-97, 2012.

[23] F. Wessel, H. Ney, “Unsupervised training of acoustic models for large vocabulary continuous speech recognition,” in IEEE Transactions on Speech and Audio Processing, 13(1), 23-3, 2005.

[24] S. Novotney, R. Schwartz, J. Ma. “Unsupervised acoustic and language model training with small amounts of labelled data,” in Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on, pp. 4297-4300. IEEE, 2009.

[25] G. Mendels, E. Cooper, V. Soto, J. Hirschberg, M. JF Gales, K. M. Knill, A. Ragni, H. Wang. “Improving speech recognition and keyword search for low resource languages using web data,” in Sixteenth Annual Conference of the International Speech Communication Association. 2015.

[26] K. Yu, M. Gales, L. Wang, P. C. Woodland. “Unsupervised training and directed manual transcription for LVCSR,” Speech Communication 52, no. 7-8 (2010): 652-663.

[27] A. Srinivasamurthy, P. Motlicek, I. Himawan, G. Szaszák, Y. Oualil, H. Helmke, “Semi-supervised learning with semantic knowledge extraction for improved speech recognition in air traffic control,” in INTERSPEECH 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm Sweden, Aug. 2017.

[28] T. Drugman, J. Pylkkönen, R. Kneser, “Active and Semi-Supervised Learning in ASR: Benefits on the Acoustic and Language Models,” in Proc. of Interspeech, 2016, pp. 2318–2322.

[29] D. Imseng, B. Potard, P. Motlicek, A. Nanchen, H. Bourlard, “Exploiting un-transcribed foreign data for speech recognition in well-resourced languages,” in Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014, pp. 2322–2326.

[30] R. Zhang, A. I. Rudnicky, “A new data selection approach for semi-supervised acoustic modeling,” in Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2006.

[31] S. Li, Y. Akita, T. Kawahara, “Semi-supervised Acoustic Model Training by Discriminative Data Selection from Multiple ASR Systems’ Hypotheses,” IEEE/ACM Transactions on Audio, Speech and Language Processing, Vol. 24, no. 9, pp. 1520–1530, Sep. 2016.

[32] R. Sarikaya, Y. Gao, M. Picheny, H. Erdogan, “Semantic confidence measurement for spoken dialog systems,” IEEE Transactions on Speech and Audio Processing, vol. 13, no. 4, pp. 534 - 545, 2005.

[33] M .Singh, Y. Oualil, D. Klakow, “Approximated and domain-adapted LSTM language models for first-pass decoding in speech recognition”, in Proceedings of the 18th Annual Conference of the International Speech Communication Association (INTERSPEECH), Stockholm, Sweden, September 2017, pp. 2720-2724.

[34] H. Adel, K. Kirchhoff, N. T. Vu, D. Telaar, T. Schultz, “Comparing approaches to convert recurrent neural networks into backoff language models for efficient decoding,” in INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14 - 18, 2014, pp. 651–655.

[35] Eurocontrol: “LINK2000+: ATC data link operational guidance in support of DLS regulation,” No 29/2009, Vol.6. 17. December 2012, online available at https://www.skybrary.aero/bookshelf/books/2383.pdf

[36] ICAO: Manual of technical provisions for the aeronautical telecommunications network (ATN), DOC9705, 2nd eds, 1999, online available at https://www.icao.int/safety/acp/repository/_%20Doc9705_ed2_1999.pdf

[37] Y. Oualil, D. Klakow, G. Szaszák, A. Srinivasamurthy, H. Helmke, P. Motlicek, “Context-aware speech recognition and understanding system for air traffic control domain”, in IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2017), Okinawa, Japan, Dec. 2017, pp. 404-408.

[38] H. Helmke, M. Slotty, M. Poiger, D. F. Herrer, O. Ohneiser et al., “Ontology for transcription of ATC speech commands of SESAR 2020 solution PJ.16-04,” in IEEE/AIAA 37th Digital Avionics Systems Conference (DASC). London, United Kingdom, 2018, to be published.

[39] M. Hössl, H. Helmke, J. Gottstein, “Why controllers seldom stick to the book and how their commands are predictable nevertheless,” in ICRAT conference, Istanbul, May. 2014.

[40] M. Kleinert, H. Helmke, G. Siol, H. Ehr, M. Finke, A. Srinivasamurthy, Y. Oualil, "Machine learning of controller command prediction models from recorded radar data and controller speech utterances," 7th SESAR Innovation Days, Belgrade, 2017.

[41] The CMU (Carnegie Mellon University) pronouncing dictionary http://www.speech.cs.cmu.edu/cgi-bin/cmudict , n.d.

[42] Eurocontrol, "All clear phraseology manual", Brussels Belgium, April 2011.

[43] F. Jelinek, R. L. Mercer, “Interpolated estimation of Markov source parameters from sparse data”,.in Proceedings of Workshop on Pattern Recognition in Practice, 1980.